Non-uniform B-spline Subdivision using Refine and Smooth

Thomas J. Cashman[1], Neil A. Dodgson[1], and Malcolm A. Sabin[2]

[1] Computer Laboratory, University of Cambridge, CB3 0FD, England
[2] Numerical Geometry Ltd, 26 Abbey Lane, Lode, Cambridge CB5 9EP, England

Abstract. Subdivision surfaces would be useful in a greater number of applications if an arbitrary-degree, non-uniform scheme existed that was a generalisation of NURBS. As a step towards building such a scheme, we investigate non-uniform analogues of the Lane-Riesenfeld 'refine and smooth' subdivision paradigm. We show that the assumptions made in constructing such an analogue are critical, and conclude that Schaefer's global knot insertion algorithm is the most promising route for further investigation in this area.

1 Background

NURBS have been an effective standard for representation of sculptured surfaces in manufacture since the late 1970's. Subdivision surfaces were also introduced at the end of the 1970's and have some significant advantages over NURBS. In particular they do not require a rigidly rectangular control grid. Both are based on the theory of B-splines and our current quad-grid subdivision surfaces can be looked on as a generalisation of a subset of NURBS.

The generalisation could be as useful in manufacture as it is in animation, where subdivision is now the de-facto standard. However, the aspects in which subdivision is only a subset are regarded as important to manufacture, and any new standard must be more compatible with the full generality of NURBS than current subdivision surfaces.

Our target is therefore a subdivision extension of NURBS in all their generality:

- non-uniform,
- rational, and
- of general degree.

Of these, the second is straightforward. Subdivision control points can be given weights and the process executed in projective space without any need for further theory.

Non-uniformity (unequal knot intervals) is not hard for low degree. Something more ambitious, assigning knot intervals to every edge on the control polyhedron, not just to every strip of polyhedron faces, was done by Sederberg et al. [10] for cubics and quadratics, and the linear case is trivial, but the handling of general degree is necessary for a subdivision method to be a generalisation of NURBS.

The work described here is an initial exploration of this target. We first address the univariate case, which we expect to map readily into tensor product surfaces. There is no point in making ad hoc rules for extraordinary vertices without having that firm foundation to build on.

In the uniform case, subdivision schemes for general degree B-splines are easily constructed and implemented using the Lane-Riesenfeld *refine and smooth* paradigm [7]. Instead of computing the new vertices directly by applying large stencils to the coarse polygon, a low degree refinement is first made locally, and then a sequence of smoothings by small, local filter operations is applied. The more smoothing steps, the higher the degree of the limit curve.

In the bivariate case, this approach gives efficiency as well as cleanness of code. Applying k 2×2 smoothing filters uses $4k$ multiplications per coordinate per new vertex; applying $\frac{1}{2}k$ 3×3 filters uses $4\frac{1}{2}k$. Applying a single $k \times k$ filter takes k^2 which is significantly more for large k.

The advantage is felt even more keenly when extraordinary vertices are introduced. Designing high-degree stencils for each valency is a non-trivial task, and designing rules for all possible combinations of extraordinary points within a large stencil is not a sensible ambition.

Although we are not addressing extraordinary points yet, the knowledge that we shall be looking at it in the future helps to steer the early work.

We want to formulate the refinement process for general degree non-uniform B-spline subdivision in terms of a composite of steps using small stencils. The archetype is the uniform refine and smooth approach.

We know exactly the complete subdivision matrices for general degree and general knot vectors. Once it has been decided what new knots are to be inserted (and this is a separate issue which we plan to address systematically), determining the new polygon is just a particular case of knot insertion. The algorithm used to insert the new knots may affect our view of the problem but is otherwise irrelevant, as the subdivision matrix is unique.

The question is how that wide-band matrix can be factorized into the narrow-band factors which correspond to the small-stencil operations that we need. This is the question addressed by the remainder of this paper.

1.1 Contents

After establishing some notation we start this exploration in Section 2, by looking at the matrices for order k and $k+2$ in the middle of a long polygon with a non-uniform knot-vector. This approach fails to find a factorisation in the general case, so we consider the particular case of Bézier end conditions in Section 3. This sub-problem is particularly important, because current subdivision schemes make a choice between

- using control points outside the surface (which leads to the edge of the surface not being particularly easily controlled), or

– using an ad hoc variation of the rules at the edge (which gives good control of the position of the edge but no control of the first derivative across it, and zero second derivative, leading to bad curvature plots).

Bézier end conditions would allow good control of both the edge itself and the tangents across it, and are described elegantly in terms of having a fully multiple knot at the end. They therefore fall naturally into a non-uniform context. However in this case, as in Section 2, we fail to find a suitable factorisation of the relevant subdivision matrices.

We discuss the reasons for this failure in Section 4 and describe a different approach from Schaefer and Goldman [4] which meets our original goals using a blossoming approach. In Section 5, we apply this technique to the Bézier end conditions of Section 3 to give a concrete example of a non-uniform subdivision matrix expressed as the product of narrow-band factors. Finally, we draw some conclusions in Section 6.

1.2 Notation

We will write τ_i for the ith knot of the original knot vector $\boldsymbol{\tau}$, and t_{2i} for the corresponding knot in the refined knot vector $\boldsymbol{t}$ ($\tau_i = t_{2i}$). t_i at odd values of i is the knot inserted between knots $\tau_{(i-1)/2}$ and $\tau_{(i+1)/2}$. As for all B-splines, we require $t_i \leq t_{i+1}$ for all i.

Let the subdivision matrix of order k (degree $k-1$) that transforms B-splines on $\boldsymbol{\tau}$ into B-splines on $\boldsymbol{t}$ be $\boldsymbol{S}^k$. Knot insertion is simply a change of basis [8], and if $B_{l,k,\boldsymbol{\gamma}}(x)$ is the lth B-spline basis function of order k on knot vector $\boldsymbol{\gamma}$, then $\boldsymbol{S}^k$ is the basis transformation matrix that gives the coordinates of each $B_{j,k,\boldsymbol{\tau}}(x)$ relative to the $B_{i,k,\boldsymbol{t}}(x)$:

$$B_{j,k,\boldsymbol{\tau}}(x) = \sum_i S^k_{ij} B_{i,k,\boldsymbol{t}}(x) \tag{1}$$

In application to subdivision, the coefficient that multiplies the jth control point in contribution to the ith new control point is therefore S^k_{ij}. (The original description of the Oslo algorithm [2] uses the notation $\alpha_{jk}(i)$ for S^k_{ij}.) Varying i for a given j produces a *mask*, and varying j for a given i produces a *stencil*.

2 Factorising Non-Uniform Knot Insertions

Uniform subdivision generalises, in the non-uniform case, to knot insertion, first published as the 'Oslo algorithm' by Cohen et al. [2], and independently as the Boehm algorithm [1]. There has been little work so far, however, on generalising Lane and Riesenfeld's work [7] to non-uniform subdivision by varying smoother mask coefficients in a similar way. Goldman et al. [5] present a construction for knots in geometric series, but do not consider the general case. Warren [11] considers a framework for non-uniform knot sequences, but limits knot vector refinement to midpoint insertion, and Gregory et al. [6] derive results for a non-uniform 'corner cutting' procedure. Neither of these approaches is based on B-spline knot insertion, however, which is required for compatibility with NURBS. In this paper, we consider two generalisations which *are* based on knot insertion: our own non-uniform refine and smooth, and Schaefer's algorithm [4].

2.1 Oslo knot insertion

We can find $\boldsymbol{S}^k$ in several equivalent ways. To begin with, since $k = 1$ gives a piecewise constant B-spline, we can see that $\boldsymbol{S}^1$ for our $\boldsymbol{\tau}$ and $\boldsymbol{t}$ is the matrix which duplicates control points (the 'refine' step of Lane-Riesenfeld):

$$\begin{bmatrix} \ddots & & & \\ & 1 & 0 & \\ & 1 & 0 & \\ & 0 & 1 & \\ & 0 & 1 & \\ & & & \ddots \end{bmatrix} \tag{2}$$

For $k > 1$, the Oslo algorithm [2] calculates the values that can be non-zero for a row of $\boldsymbol{S}^k$ using a triangular scheme. The sum of the values in each row of $\boldsymbol{S}^k$ is always 1, and each row is therefore 'smeared out' as k increases — echoing, in a discrete way, the continuous convolution of B-spline basis functions. There is a triangle generated by the Oslo algorithm for each row of $\boldsymbol{S}^k$, with an apex at each '1' of $\boldsymbol{S}^1$.

2.2 Understanding uniform refine and smooth

We want to formulate the properties of a *uniform* refine and smooth factorisation in terms of the knot insertion matrices $\boldsymbol{S}^k$, and then generalise to a non-uniform factorisation with similar properties. To do so, we need to look at the following appealing property of the Lane-Riesenfeld algorithm.

– After d smoothing steps, the points are the control polygon for a uniform B-spline of degree d which matches the B-spline of the same degree (with knot intervals doubled) defined on the original polygon.

This property cannot be generalised in terms of constant knot vectors $\boldsymbol{\tau}$ and $\boldsymbol{t}$. To see this, consider a single knot interval $[\tau_i \; \tau_{i+1}]$. The knot t_{2i+1} is inserted here and the relevant part of $\boldsymbol{S}^1$ (for *constant* basis functions) is $\left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)$, duplicating the control point which corresponds to the original interval. The result of smoothing these values in any ratio is therefore to return the same original point. *Linear* basis functions, however, have a support of two knot intervals, so within $[\tau_i \; \tau_{i+1}]$ there is only one new basis function and the relevant part of $\boldsymbol{S}^2$ is $(\frac{1}{2} \; \frac{1}{2})$, taking a mean of adjacent points.

This problem arises because where k is even, control points correspond to knot *values*. Where k is odd, they are paired with knot *intervals*. We therefore cannot hope to compare $\boldsymbol{S}^j$ with $\boldsymbol{S}^k$ in the non-uniform case if $|j-k|$ is odd. To do so, we would need to either change $\boldsymbol{\tau}$ and $\boldsymbol{t}$ or violate symmetry.

Bearing this in mind, we might list properties of a uniform refine and smooth factorisation as follows. We take $\theta = 1 + (k+1 \bmod 2)$, so $\theta = 1$ where k is odd, and 2 where k is even.

A. For any k, there are smoothing matrices $\boldsymbol{M}^\theta, \ldots, \boldsymbol{M}^{k-1}$ such that $\boldsymbol{S}^k = \boldsymbol{M}^{k-1}\boldsymbol{M}^{k-2}\ldots\boldsymbol{M}^\theta\boldsymbol{S}^\theta$.
B. Furthermore, for each $\kappa < k$ such that $k-\kappa$ is even, $\boldsymbol{M}^{\kappa-1}\ldots\boldsymbol{M}^\theta\boldsymbol{S}^\theta = \boldsymbol{S}^\kappa$.
C. Each $\boldsymbol{M}$ is a band matrix of bandwidth 2.
D. Each row of each $\boldsymbol{M}$ performs $\frac{1}{2}, \frac{1}{2}$ averaging.

No factorisation for non-uniform knot insertion can hold all four of these properties, so we are looking for a generalisation which maintains a chosen subset. In particular, property A makes the factorisation useful and property C is important for the benefits of locality discussed in Section 1. In Section 4, we will consider Schaefer's generalisation [4] of refine and smooth which holds just these two of the four. In this section, we consider a non-uniform analogue of the Lane-Riesenfeld approach which also maintains property B.

2.3 Our generalised refine and smooth

Since our non-uniform refine and smooth will not hold property D, we allow each row $\boldsymbol{M}_i^k$ of the smoothing matrices to hold a different ratio. These ratios may differ within the same matrix and between different smoothing matrices — we only require that the weights in each row still sum to 1. Taking affine combinations of points is necessary in order to retain invariance under solid-body (and affine) transformations.

We can also consider the product of two smoothing matrices $\boldsymbol{M}^{k+1}\boldsymbol{M}^k$ as a single smoothing matrix which takes a weighted mean of three rows. If the weights in the ith row of the smoother matrix that gives $\boldsymbol{S}^{d+2}$ from $\boldsymbol{S}^d$ are α_i^d, β_i^d and $1-(\alpha_i^d+\beta_i^d)$, then we require

$$\begin{aligned} &\alpha_i^d\boldsymbol{S}_i^d + \beta_i^d\boldsymbol{S}_{i+1}^d + (1-\alpha_i^d-\beta_i^d)\boldsymbol{S}_{i+2}^d = \boldsymbol{S}_i^{d+2} \\ \text{i.e.}\qquad &\alpha_i^d(\boldsymbol{S}_i^d - \boldsymbol{S}_{i+2}^d) + \beta_i^d(\boldsymbol{S}_{i+1}^d - \boldsymbol{S}_{i+2}^d) = \boldsymbol{S}_i^{d+2} - \boldsymbol{S}_{i+2}^d \end{aligned} \tag{3}$$

Or expressed as a system of equations (which must be satisfied for all j),

$$\begin{pmatrix} \alpha_i^d & \beta_i^d \end{pmatrix} \begin{pmatrix} \boldsymbol{S}_i^d - \boldsymbol{S}_{i+2}^d \\ \boldsymbol{S}_{i+1}^d - \boldsymbol{S}_{i+2}^d \end{pmatrix} = (\boldsymbol{S}_i^{d+2} - \boldsymbol{S}_{i+2}^d). \tag{4}$$

We only want to look at every second d. If k is even, $d \in \{2, 4, \ldots, k\}$ and if k is odd, $d \in \{1, 3, \ldots, k\}$. This means that the 'refine' step of our generalisation is $\boldsymbol{S}^\theta$ from property A. If the resulting systems of equations (4), one for each value of j, has a solution for α_i^d and β_i^d for every i and relevant d, then we will say that the knot vector refinement $\boldsymbol{t}$ of $\boldsymbol{\tau}$ has a refine and smooth formulation.

2.4 Two stencils to consider

In order to characterise the behaviour of our refine and smooth factorisation, we now need to establish the values of i we need to consider in the systems (4). Each system sets up a correspondence between a set of four rows: three in $\boldsymbol{S}^d$ and another in $\boldsymbol{S}^{d+2}$. There is no need to examine two such sets if there is an isomorphism between the two.

As our $\boldsymbol{\tau}$ and $\boldsymbol{t}$ perform binary subdivision, there are only two types of stencil appearing in each $\boldsymbol{S}^k$, and there are therefore two differing sets of rows to consider. At $\boldsymbol{S}^1$, the two groups of three rows, which are both smoothed to obtain a row of $\boldsymbol{S}^3$, are

$$\begin{matrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{matrix} \quad \text{and} \quad \begin{matrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{matrix}\,. \tag{5}$$

Every other collection of three rows is isomorphic to one of these cases, where the required mapping involves just a shift and subscript-rewriting. We can obtain the row entries for both cases (5) using the Oslo algorithm [2] and check every resulting linear system (4) to a given depth k.

The number of equations to be satisfied in a system (4) depends on the width of the stencils in the relevant rows. These stencils grow wider with k, so we expect the systems to become more constrained as k grows. If k is large enough, we may require additional constraints on $\boldsymbol{\tau}$ and $\boldsymbol{t}$ in order to satisfy the systems, as each smoother has just two degrees of freedom.

2.5 Factorisable knot insertions

We now have a target factorisation which expresses a knot insertion matrix $\boldsymbol{S}^k$ as the product of a refinement matrix and $\lfloor \frac{k-1}{2} \rfloor$ smoothing matrices. To establish for which knot insertion cases this factorisation exists, we used MATLAB to analyze the systems (4) in the two cases (5) and find that

- for $k \leq 5$, our refine and smooth formulation exists for every possible knot insertion.
- for $k \geq 6$ (degree at least quintic), equations limit the possible configurations for $\boldsymbol{\tau}$ and $\boldsymbol{t}$.

We can find an equation that relates knots in both $\boldsymbol{\tau}$ and $\boldsymbol{t}$, but the analysis is more manageable when considering specific configurations for $\boldsymbol{\tau}$. For the rest of this section we will be considering the case for $k = 6$, a quintic B-spline, where we take $\boldsymbol{\tau}$ to be uniform, and then a sequence in geometric series. The results for these specific $\boldsymbol{\tau}$ provide some insight into the general behaviour.

Knot insertion on a uniform knot vector. For a uniform knot vector, we set $\tau_i = i$. Instead of writing down a different equation for different positions, we will give a single equation that characterises the constraints for a whole knot vector. To do so, we write $r_i = (t_{2i+1} - \tau_i)/(\tau_{i+1} - \tau_i)$. Uniform $\boldsymbol{\tau}$ then gives

$r_i = t_{2i+1} - i$, and we find that our refine and smooth formulation requires for every i

$$\begin{aligned} 0 = 4r_i^3 &- 3r_i^2(r_{i-1} + 1 + r_{i+1}) \\ &+ 2r_i(2r_{i-1} + 1 + r_{i-1}r_{i+1}) - r_{i-1}r_{i+1} - 2r_{i-1} \end{aligned} \tag{6}$$

Figure 1 shows a contour plot of r_{i+1} in terms of r_{i-1} and r_i. For $\boldsymbol{\tau}$ and $\boldsymbol{t}$ to have the format described in Section 1.2, we require $0 \leq r_{i+1} \leq 1$. Figure 1 therefore shows the boundary for values of r_{i-1} and r_i which lead to viable values for r_{i+1}. For a refined knot vector with more than 7 knots[3], we must also consider the equation that constrains r_i, $r_i + 1$ and $r_i + 2$, which is identical to (6) with $i+1$ for i. In a similar fashion, we can consider the effect of r_{i-1} and r_i on r_{i-2}. The area of possible (r_{i-1}, r_i) values shrinks, the more knots there are in $\boldsymbol{t}$.

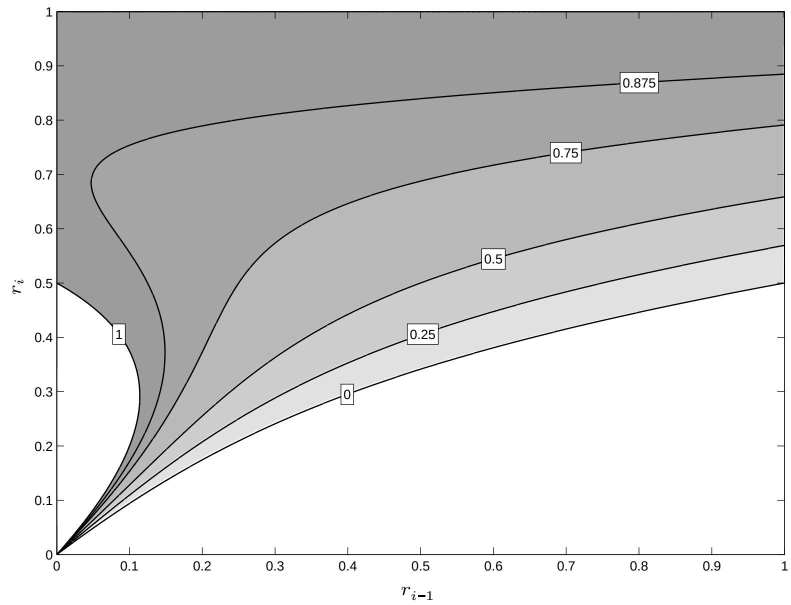


Fig. 1: The value of r_{i+1} to produce a knot insertion into a uniform knot vector with a refine and smooth formulation, when constrained by r_{i-1} and r_i. The shaded region gives $0 \leq r_{i+1} \leq 1$ and therefore shows the valid values for (r_{i-1}, r_i).

We already know, though, that $r_i = \frac{1}{2}$ for all i is a refine and smooth-factorisable knot insertion, irrespective of the length of $\boldsymbol{t}$. This is uniform knot insertion into a uniform knot vector — exactly Lane-Riesenfeld subdivision [7]. In fact, it is simple to verify from (6) that $r_i = v$ for all i is also a refine

[3] 3 inserted knots and the 4 original uniform knots

and smooth knot insertion that preserves property B, for any $0 \leq v \leq 1$. We conjecture that this will be true for a uniform knot vector at any degree.

Knot insertion on a knot vector in geometric series. We now turn to a knot vector in geometric series with ratio 2, by setting $\tau_i = 2^i$. We can still completely characterise the constraints on $\boldsymbol{t}$ using an equation in terms of the ratios r_i (now $r_i = (t_{2i+1}/2^i) - 1$), but the equation to be satisfied is not as neat as (6):

$$\begin{aligned} 0 &= 4369r_i^3 - 3r_i^2(91r_{i-1} - 334 + 1456r_{i+1}) \\ &\quad + r_i(22r_{i-1} + 5 - 1008r_{i+1} + 272r_{i-1}r_{i+1}) - 16r_{i-1}r_{i+1} - 5r_{i-1} \end{aligned} \tag{7}$$

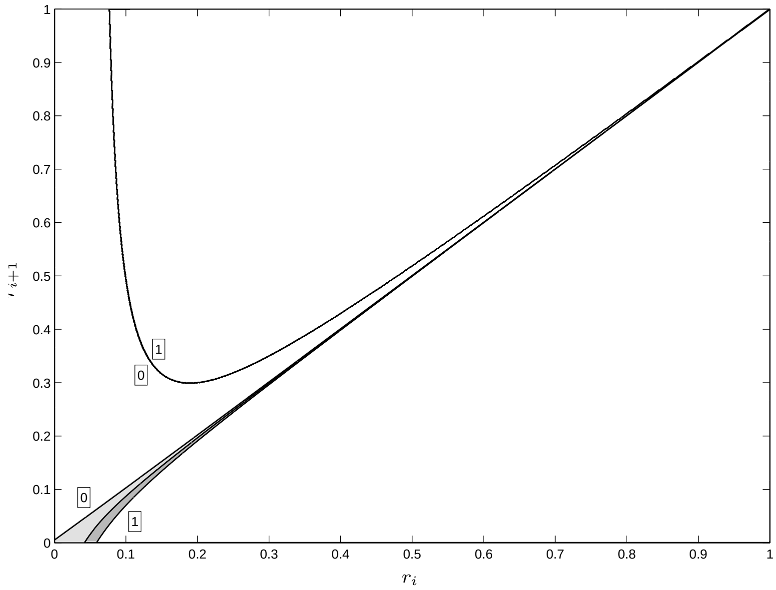


Fig. 2: The value of r_{i-1} to produce a knot insertion into a geometric series knot vector with a refine and smooth formulation, when constrained by r_i and r_{i+1}

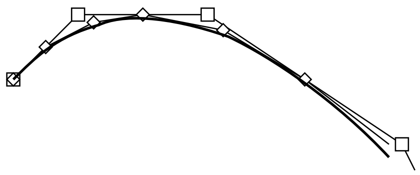
Figure 2 shows the contour plot of r_{i-1} in terms of r_i and r_{i+1}, as this puts far greater constraints on the possible values of (r_i, r_{i+1}) than the value of r_{i+2} does. We can see that in the geometric case, the two degrees of freedom in assigning (r_i, r_{i+1}) are nearly reduced to just one, as little deviation is possible away from the solution $r_i = r_{i+1}$. However, in a surprising similarity with the uniform case, we can still show that $r_i = v$ for all i is a refine and smooth knot insertion, for any $0 \leq v \leq 1$, and regardless of the length of $\boldsymbol{t}$.

This is immediately useful when inserting into a geometric series knot vector, as we can choose v such that the refined knot vector is also in geometric series (here, we would have $v = \sqrt{2} - 1$). The refined knot vector can then be mapped onto the original using just scales and shifts in parameter, and so it is possible to subdivide the *refined* knot vector (using refine and smooth) with any $0 \leq v \leq 1$ as well. This was first shown by Goldman and Warren [5].

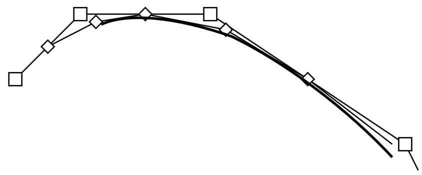
3 Bézier End Conditions

These constraints on factorisable knot insertions make our refine and smooth formulation unusable in practice. We consider the general problem again in Section 4, but first we look at the same approach in the light of a different context; that of Bézier end conditions.

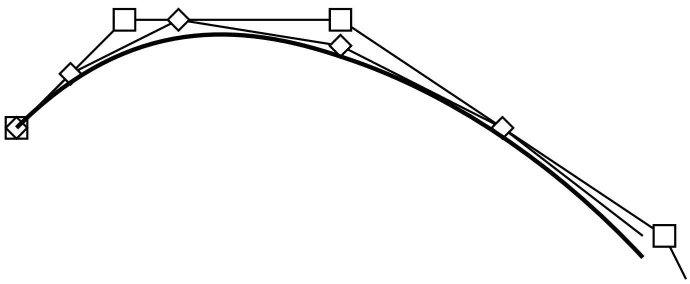
If the first and last knots in an otherwise uniform knot-vector have multiplicity of at least the degree of the B-spline, the ends of the curve behave in a very similar way to those of a Bézier curve. The end control point determines the position of the end of the curve, the end two determine the first derivative, the end three the second derivative and so on. In fact if there are no internal knots, the curve is a Bézier curve. Because this scenario is just a particular case of the general non-uniform B-spline, all the properties of affine invariance and containment in the convex hull are retained.



(a) Ad hoc end with zero curvature



(b) Truncated polygon



(c) Bézier end-condition

Fig. 3: Cubic limit curves with their control polygons

This nice behaviour should be contrasted with the ways in which subdivision curves are usually terminated;

– truncation, where the only control points after one refinement are those which are well-defined by the same rules as are used in the interior of the curve. The polygon gets shorter at every refinement, and the end of the limit curve can only be precisely positioned by solving a linear system to determine where the outer control points should be.
– The usual ad hoc fixup, where control points for which the general rules cannot be applied because their antecedents do not exist, are given special rules. In particular in the cubic case, only the end control point after refinement is undefined, and it is typically positioned at the end point before refinement. This enables easy positioning of the end of the limit curve, but results in the second derivative being zero there.

We would like to have a systematic approach which can be used to either build explicit subdivision matrices for Bézier end conditions, or, hopefully, to be just a particular case within the refine and smooth paradigm.

3.1 Systematic development

We are looking at setting, for example, $\tau_i = 0$ for $i \leq 0$ and $\tau_i = 2i, t_i = i$ for $i \geq 1$. Examples of the resulting $\boldsymbol{S}^k$ are given for $k = 2 \ldots 5$ in Fig. 4, and a method for computation of these specific $\boldsymbol{S}^k$ is described in [3].

Linear, $k = 2$

$$\frac{1}{2}\begin{bmatrix} 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & 1 & \\ & & & \ddots \end{bmatrix}$$

Quadratic, $k = 3$

$$\frac{1}{4}\begin{bmatrix} 4 & & & & \\ 2 & 2 & & & \\ & 3 & 1 & & \\ & 1 & 3 & & \\ & & 3 & 1 & \\ & & 1 & 3 & \\ & & & & \ddots \end{bmatrix}$$

Cubic, $k = 4$

$$\frac{1}{16}\begin{bmatrix} 16 & & & & & \\ 8 & 8 & & & & \\ & 12 & 4 & & & \\ & 3 & 11 & 2 & & \\ & & 8 & 8 & & \\ & & 2 & 12 & 2 & \\ & & & & & \ddots \end{bmatrix}$$

Quartic, $k = 5$

$$\frac{1}{48}\begin{bmatrix} 48 & & & & & & & \\ 24 & 24 & & & & & & \\ & 36 & 12 & & & & & \\ & 9 & 33 & 6 & & & & \\ & & 20 & 25 & 3 & & & \\ & & 4 & 29 & 15 & & & \\ & & & 15 & 30 & 3 & & \\ & & & 3 & 30 & 15 & & \\ & & & & & & & \ddots \end{bmatrix}$$

Fig. 4: The knot insertion matrices $\boldsymbol{S}^k$ for $k = 2 \ldots 5$ and Bézier end conditions

In the uniform case (with cyclic data from a closed polygon) we can determine $\boldsymbol{S}^{k+1}$ from $\boldsymbol{S}^k$ merely by multiplying by the circulant smoothing matrix

$$\begin{bmatrix} \ddots & & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & & & & & \ddots \end{bmatrix}$$

We are interested in seeing whether we can derive $\boldsymbol{S}^{k+1}$ from $\boldsymbol{S}^k$ in a similar manner in the Bézier end condition case. For the reasons discussed in Section 1, we would still like the matrix that multiplies $\boldsymbol{S}^k$ to have a narrow band. In order to derive this multiplier, we observe three interesting properties from the examples in Fig. 4:

1. that the top left $k \times k$ square of $\boldsymbol{S}^k$ is identical to the top left corner of $\boldsymbol{S}^{k+1}$ (remember the global scaling factor on each matrix).
2. that beyond the $(k-1)$th column the matrix is regular.
3. that beyond the $(2k-4)$th row the matrix is regular.

The interesting questions are

- whether these matrices can be determined systematically other than by carrying out the full knot insertion process for each degree.
- whether the refine and smooth paradigm, which is so elegant for the implementation of high degree uniform B-splines as subdivision curves, can be extended to cover this case.

Now if $\boldsymbol{S}^{k+1} = \boldsymbol{M}\boldsymbol{S}^k$ we would expect to be able to derive $\boldsymbol{M}$ from the product $\boldsymbol{S}^{k+1}(\boldsymbol{S}^k)^{-1}$, but this is not straightforward. Each of the $\boldsymbol{S}^k$s is about twice as high as it is wide, and therefore inversion is not a well defined process. In fact $\boldsymbol{M}$ has about twice the number of elements as the number of conditions we are trying to satisfy, and so it is heavily underdetermined.

We are actually looking for a sparse structure, and just using, for example, the Penrose pseudo-inverse does not give the sparsity that would make the factorisation useful.

Property 1, above, suggests that the top left corner of $\boldsymbol{M}$ should be a unit matrix, and that the remainder of the first $k-1$ rows should be zero. Properties 2 and 3 suggest that beyond the $(2k-4)$th row, the pattern of uniform smoothing should be present. The question is what happens in between. We look at a couple of example cases.

Quadratic to Cubic

$$\left[\begin{array}{ccc|ccc} 16 & & & & & \\ 8 & 8 & & & & \\ & 12 & 4 & & & \\ \hline & 3 & 11 & 2 & & \\ & & 8 & 8 & & \\ & & 2 & 12 & 2 & \\ & & & & & \ddots \end{array}\right] = \left[\begin{array}{ccc|ccc} 4 & & & & & \\ & 4 & & & & \\ & & 4 & & & \\ \hline -1 & 2 & -1 & 2 & 2 & \\ & & & & 2 & 2 \\ & & & & & 2 \\ & & & & & & \ddots \end{array}\right] \left[\begin{array}{ccc|cc} 4 & & & & \\ 2 & 2 & & & \\ & 3 & 1 & & \\ \hline & 1 & 3 & & \\ & & 3 & 1 & \\ & & 1 & 3 & \\ & & & & \ddots \end{array}\right]$$

Cubic to Quartic

$$
\left[\begin{array}{cccc|ccc}
48 & & & & & & \\
24 & 24 & & & & & \\
 & 36 & 12 & & & & \\
 & 9 & 33 & 6 & & & \\
\hline
 & & 20 & 25 & 3 & & \\
 & & 4 & 29 & 15 & & \\
 & & & 15 & 30 & 3 & \\
 & & & 3 & 30 & 15 & \\
 & & & & & & \ddots
\end{array}\right]
$$

$$
= \frac{1}{8}
\left[\begin{array}{cccc|ccccc}
24 & & & & & & & & \\
 & 24 & & & & & & & \\
 & & 24 & & & & & & \\
 & & & 24 & & & & & \\
\hline
45 & -90 & 65 & -20 & 12 & 12 & & & \\
9 & -18 & 13 & -4 & & 12 & 12 & & \\
 & & & & & & 12 & 12 & \\
 & & & & & & & 12 & \\
 & & & & & & & & \ddots
\end{array}\right]
\left[\begin{array}{cccc|ccc}
16 & & & & & & \\
8 & 8 & & & & & \\
 & 12 & 4 & & & & \\
 & 3 & 11 & 2 & & & \\
\hline
 & & 8 & 8 & & & \\
 & & 2 & 12 & 2 & & \\
 & & & 8 & 8 & & \\
 & & & 2 & 12 & 2 & \\
 & & & & & & \ddots
\end{array}\right]
$$

3.2 The need for a different approach

For Bézier end conditions, the matrix $\boldsymbol{M}$ by which we multiply the order k subdivision matrix to get the order $k+1$ subdivision matrix does not appear to have a derivable narrow band in the region where the two kinds of regularity meet. These $\boldsymbol{M}$ matrices are therefore performing non-local operations, and so are not useful in the way described in Section 1.

For the property-B-preserving general case explored in Section 2, the result is even worse. Although we can enforce local smoothing operations (property C), we must pay for the privilege on two counts. Firstly, we create **knot dependencies**; in the quintic case, positioning two inserted knots determines every other inserted knot. Secondly, the resulting structure is **restricted**; not all knot insertions are factorisable in this way.

We want the position of inserted knots to be a design decision, influenced by factors such as convergence to uniformity. Instead we have a system where the knot insertion machinery prescribes the position of inserted knots, and this is not a viable route for further investigation.

4 Schaefer's Knot Insertion Algorithm

The work of Goldman and Schaefer [4] throws new light on these results by showing the impact of retaining property B described in Section 2. In this section, we describe *Schaefer's knot insertion* algorithm, which constructs an analogue of the Lane-Riesenfeld algorithm for arbitrary knot vectors without suffering from the restrictions, loss of locality or knot dependencies seen in Sections 2 and 3. The algorithm achieves this by seeking to maintain only properties A and C of uniform refine and smooth in the non-uniform context.

Schaefer's algorithm has a direct blossoming [9] proof and derivation. It is best illustrated using diagrams of the following format (this example is for a quadratic spline):

$$\begin{tikzcd}[column sep=small] & f(\alpha,t_2) & \\ f(t_1,t_2) \arrow[ur] & & f(t_2,t_3) \arrow[ul] \end{tikzcd} \quad \text{which we abbreviate as} \quad \begin{tikzcd}[column sep=small] & \alpha\, t_2 & \\ t_1\, t_2 \arrow[ur] & & t_2\, t_3 \arrow[ul] \end{tikzcd}$$

This diagram represents the affine combination of blossoms

$$f(\alpha, t_2) = \frac{t_3 - \alpha}{t_3 - t_1} f(t_1, t_2) + \frac{\alpha - t_1}{t_3 - t_1} f(t_2, t_3) \tag{8}$$

We recall that B-spline control points are given by blossoms of the form $f(\tau_i, \ldots, \tau_{i+k-2})$ for adjacent knots. Within this scheme, linear Lane-Riesenfeld subdivision is represented by the diagram

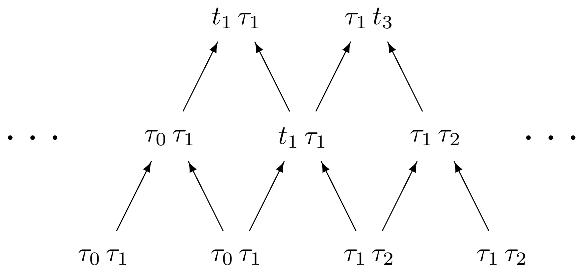
$$\begin{tikzcd}[column sep=small] & & 0 & & \frac{1}{2} & & 1 & & \\ \cdots & 0 \arrow[ur] & & 0 \arrow[ul] \arrow[ur] & & 1 \arrow[ul] \arrow[ur] & & 1 \arrow[ul] & \cdots \end{tikzcd}$$

Blossoms are repeated at the bottom level of the diagram, as duplicating control points is the first step in Lane-Riesenfeld refine and smooth. The following diagram subsequently employs two smoothing steps, for quadratic Lane-Riesenfeld subdivision:

$$\begin{tikzcd}[column sep=small] & & & \frac{1}{2}\, 1 & & 1\, 1\frac{1}{2} & & & \\ \cdots & & 0\, 1 \arrow[ur] & & 1\, 1 \arrow[ul] \arrow[ur] & & 1\, 2 \arrow[ul] & & \cdots \\ & 0\, 1 \arrow[ur] & & 0\, 1 \arrow[ul] \arrow[ur] & & 1\, 2 \arrow[ul] \arrow[ur] & & 1\, 2 \arrow[ul] & \end{tikzcd}$$

At the top level of the diagram, blossoms are produced with adjacent knots in a *refined* knot vector, with half the spacing of the original knots.

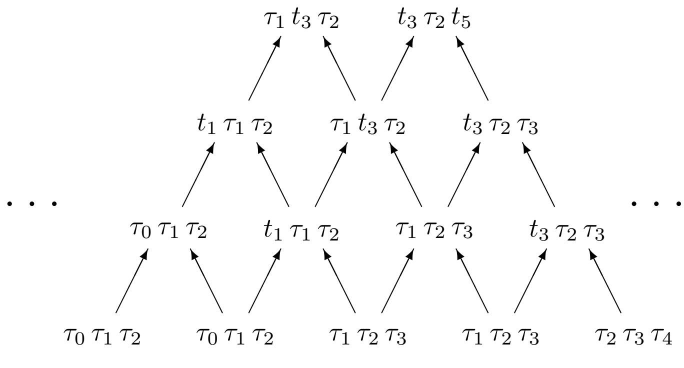
In both the linear and quadratic cases, Lane-Riesenfeld subdivision can be extended to non-uniform knot vectors simply by substituting the non-uniform values into the blossom diagrams above. For degree greater than quadratic, however, it becomes hard to prove properties of Lane-Riesenfeld refine and smooth with these diagrams. Schaefer's knot insertion, by contrast, can be readily proved with blossoming diagrams for arbitrary degree. This is the diagram which represents Schaefer's knot insertion for quadratics:



There are several important properties of Schaefer's knot insertion factorisation:

- Knots are inserted "as soon as possible". The Lane-Riesenfeld algorithm calculates all the required points in the final smoothing step, whereas Schaefer's algorithm calculates half the points in the penultimate smoothing, and the remaining half in the final step.
- Half of the affine combinations in any given row simply select one of the blossoms from the row below. In this sense, Schaefer's knot insertion is faster than Lane-Riesenfeld, as it only performs half the work.
- Schaefer's knot insertion does not specialise to the Lane-Riesenfeld construction in the case where knot spacings are uniform. Goldman [4] presents two algorithms for cubic knot insertion which do have this property, but have not been extended to general degree.
- The central blossom value in the above diagram could equally have been $\tau_1\,t_3$, and Schaefer's algorithm makes an arbitrary choice between these two asymmetric options.

A final property of Schaefer's algorithm, which is attractive for the sake of understanding, is that the diagram representing the cubic algorithm is readily derived from the quadratic diagram by simply appending blossom arguments. To make this clear, here is the diagram for the cubic case:



5 A Factorisation for Bézier End Conditions

We can now apply Schaefer's algorithm from Section 4 to the Bézier end conditions considered in Section 3. This illustrates that Schaefer's algorithm provides a framework for implementing subdivision curves and surfaces, with Bézier end conditions, using a refine and smooth method. Here we provide the factorisations for linear, quadratic and cubic curves. Since Schaefer's algorithm does not reduce to Lane-Riesenfeld for uniform knots, the smoothing matrices do not compute arithmetic means in the regular regions.

Linear

$$\frac{1}{2}\begin{bmatrix} 2 & & \\ 1 & 1 & \\ & 2 & \\ & 1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & 2 & & & \\ & 1 & 1 & & \\ & & 0 & 2 & \\ & & & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & & 1 \end{bmatrix}$$

Quadratic

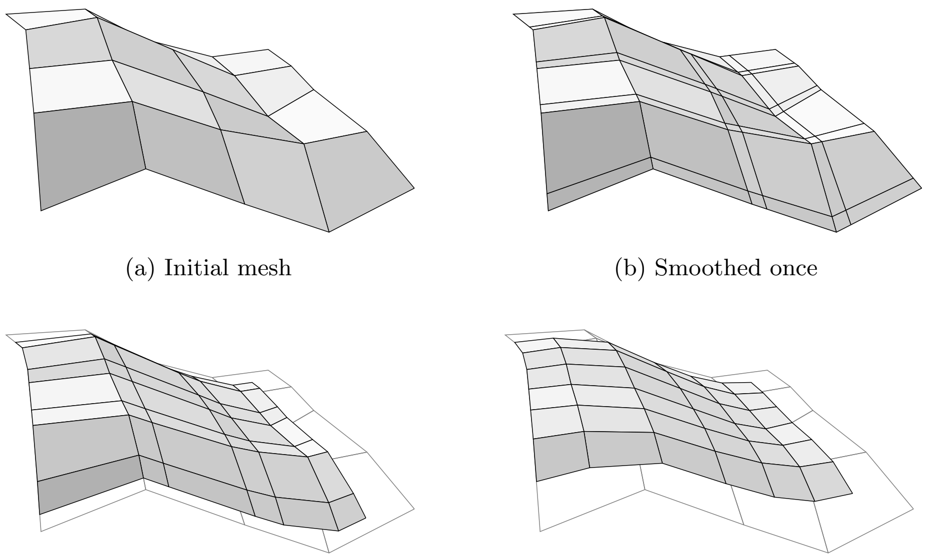
$$\frac{1}{4}\begin{bmatrix} 4 & & & \\ 2 & 2 & & \\ & 3 & 1 & \\ & 1 & 3 & \\ & & 3 & 1 \\ & & 1 & 3 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 0 & 6 & & & & & \\ & 3 & 3 & & & & \\ & & 0 & 6 & & & \\ & & & 2 & 4 & & \\ & & & & 0 & 6 & \\ & & & & & 2 & 4 \end{bmatrix}\frac{1}{4}\begin{bmatrix} 0 & 4 & & & & & & \\ & 4 & 0 & & & & & \\ & & 0 & 4 & & & & \\ & & & 3 & 1 & & & \\ & & & & 0 & 4 & & \\ & & & & & 3 & 1 & \\ & & & & & & 0 & 4 \end{bmatrix}\begin{bmatrix} 1 & & & \\ 1 & & & \\ & 1 & & \\ & 1 & & \\ & & 1 & \\ & & 1 & \\ & & & 1 \\ & & & 1 \end{bmatrix}$$

Cubic

$$\frac{1}{16}\begin{bmatrix} 16 & & & & \\ 8 & 8 & & & \\ & 12 & 4 & & \\ & 3 & 11 & 2 & \\ & & 8 & 8 & \\ & & 2 & 12 & 2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 0 & 4 & & & & & \\ & 2 & 2 & & & & \\ & & 0 & 4 & & & \\ & & & 1 & 3 & & \\ & & & & 0 & 4 & \\ & & & & & 1 & 3 \end{bmatrix}\frac{1}{20}\begin{bmatrix} 0 & 20 & & & & & & \\ & 20 & 0 & & & & & \\ & & 0 & 20 & & & & \\ & & & 15 & 5 & & & \\ & & & & 0 & 20 & & \\ & & & & & 12 & 8 & \\ & & & & & & 0 & 20 \end{bmatrix}$$

$$\times\frac{1}{6}\begin{bmatrix} 0 & 6 & & & & & & & \\ & 6 & 0 & & & & & & \\ & & 0 & 6 & & & & & \\ & & & 6 & 0 & & & & \\ & & & & 0 & 6 & & & \\ & & & & & 5 & 1 & & \\ & & & & & & 0 & 6 & \\ & & & & & & & 5 & 1 \end{bmatrix}\begin{bmatrix} 1 & & & & \\ 1 & & & & \\ & 1 & & & \\ & 1 & & & \\ & & 1 & & \\ & & 1 & & \\ & & & 1 & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

Figure 5 shows this factorisation in one direction of a tensor-product application of Schaefer's algorithm. The orthogonal direction uses the factorisation for uniform subdivision.



(a) Initial mesh (b) Smoothed once

(c) Smoothed twice (d) Refined mesh

Fig. 5: A non-uniform bicubic subdivision step using Schaefer's algorithm. The left-hand edge has a Bézier end condition

6 Conclusions

– A refine and smooth factorisation of knot insertion matrices is useful for implementation, and may prove a useful step towards non-uniform subdivision schemes of general degree.

– The definition of "refine and smooth factorisation" can make a marked difference to what is possible and what the factorisation looks like.

– Schaefer's algorithm performs non-uniform knot insertion for any degree, using a refine and smooth factorisation, and is easily derived using blossoming. The factorisation may prove useful conceptually as well as in implementation, and merits further investigation.

References

1. Boehm, W.: Inserting new knots into B-spline curves. Computer-Aided Design **12**(4) (1980) 199–201
2. Cohen, E., Lyche, T., Riesenfeld, R.: Discrete *B*-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics. Computer Graphics and Image Processing **14**(2) (1980) 87–111
3. Gasciola, G., Romani, L.: A general matrix representation for non-uniform B-spline subdivision with boundary control. Draft paper (2006)
4. Goldman, R., Schaefer S.: Global Knot Insertion Algorithms. Presentation at the University of Kaiserslautern (17th January, 2007)
5. Goldman, R., Warren, J.: An extension of Chaiken's algorithm to B-spline curves with knots in geometric progression. CVGIP: Graphical Models and Image Processing **55**(1) (1993) 58–62
6. Gregory, J., Qu, R.: Nonuniform corner cutting. Computer Aided Geometric Design **13**(8) (1996) 763–772
7. Lane, J., Riesenfeld, R.: A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **2**(1) (1980) 35–46
8. Lyche, T., Morken, K.: Making the OSLO Algorithm More Efficient. SIAM Journal on Numerical Analysis **23**(3) (June 1986) 663–675
9. Ramshaw, L.: Blossoming: A Connect-the-Dots Approach to Splines. Technical Report 19, Digital Systems Research Center (1987)
10. Sederberg, T., Zheng, J., Sewell, D., Sabin, M.: Non-Uniform Recursive Subdivision Surfaces. Proceedings of the 25th annual conference on Computer Graphics and Interactive Techniques (1998) 387–394
11. Warren, J.: Binary subdivision schemes for functions over irregular knot sequences. In Dæhlen, M., Lyche, T., Schumaker, L.L., eds.: Mathematical Methods for Curves and Surfaces. Vanderbilt U.P. (1995) 543–562 ISBN 0-8265-1268-2.